



**STANFORD UNIVERSITY**

STANFORD, CALIFORNIA 94305-2140

DONALD E. KNUTH  
Professor Emeritus of The Art of  
Computer Programming  
Department of Computer Science  
Telephone [415] 723-4367

February 23, 1994

Commissioner of Patents and Trademarks  
Box 4  
Patent and Trademark Office  
Washington, DC 20231

Dear Commissioner:

Along with many other computer scientists, I would like to ask you to reconsider the current policy of giving patents for computational processes. I find a considerable anxiety throughout the community of practicing computer scientists that decisions by the patent courts and the Patent and Trademark Office are making life much more difficult for programmers.

In the period 1945–1980, it was generally believed that patent law did not pertain to software. However, it now appears that some people have received patents for algorithms of practical importance—e.g., Lempel-Ziv compression and RSA public key encryption—and are now legally preventing other programmers from using those algorithms.

This is a serious change from the previous policy under which the computer revolution became possible, and I fear this change will be harmful for society. It certainly would have had profoundly negative effect on my own work: For example, I developed software called  $\text{\TeX}$  that is now used to produce more than 90% of all books and journals in mathematics and physics and to produce hundreds of thousands of technical reports in all scientific disciplines. If software patents had been commonplace in 1980, I would not have been able to create such a system, nor would I probably have ever thought of doing it, nor can I imagine anyone else doing so.

I am told that the courts are trying to make a distinction between mathematical algorithms and nonmathematical algorithms. To a computer scientist, this makes no sense, because every algorithm is as mathematical as anything could be. An algorithm is an abstract concept unrelated to physical laws of the universe.

Nor is it possible to distinguish between “numerical” and “nonnumerical” algorithms, as if numbers were somehow different from other kinds of precise information. All data are numbers, and all numbers are data. Mathematicians work much more with symbolic entities than with numbers.

Therefore the idea of passing laws that say some kinds of algorithms belong to mathematics and some do not strikes me as absurd as the 19th century attempts of the Indiana legislature to pass a law that the ratio of a circle's circumference to its diameter is exactly 3, not approximately 3.1416. It's like the medieval church ruling that the sun revolves about the earth. Man-made laws can be significantly helpful but not when they contradict fundamental truths.

Congress wisely decided long ago that mathematical things cannot be patented. Surely nobody could apply mathematics if it were necessary to pay a license fee whenever the theorem of Pythagoras is employed. The basic algorithmic ideas that people are now rushing to patent are so fundamental, the result threatens to be like what would happen if we allowed authors to have patents on individual words and concepts. Novelists or journalists would be unable to write stories unless their publishers had permission from the owners of the words. Algorithms are exactly as basic to software as words are to writers, because they are the fundamental building blocks needed to make interesting products. What would happen if individual lawyers could patent their methods of defense, or if Supreme Court justices could patent their precedents?

I realize that the patent courts try their best to serve society when they formulate patent law. The Patent Office has fulfilled this mission admirably with respect to aspects of technology that involve concrete laws of physics rather than abstract laws of thought. I myself have a few patents on hardware devices. But I strongly believe that the recent trend to patenting algorithms is of benefit only to a very small number of attorneys and inventors, while it is seriously harmful to the vast majority of people who want to do useful things with computers.

When I think of the computer programs I require daily to get my own work done, I cannot help but realize that none of them would exist today if software patents had been prevalent in the 1960s and 1970s. Changing the rules now will have the effect of freezing progress at essentially its current level. If present trends continue, the only recourse available to the majority of America's brilliant software developers will be to give up software or to emigrate. The U.S.A. will soon lose its dominant position.

Please do what you can to reverse this alarming trend. There are far better ways to protect the intellectual property rights of software developers than to take away their right to use fundamental building blocks.

Sincerely,



Donald E. Knuth  
Professor